Merrimac – Supercomputing with Streams

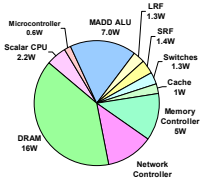
http://merrimac.stanford.edu
mattan.erez@stanford.edu

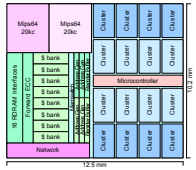
Mattan Erez, Jung Ho Ahn, Nuwan Jayasena, Timothy J. Knight, Abhishek Das, Francois Labonte, Jayanth Gummaraju

The Merrimac streaming supercomputer project aims to develop a scientific computer that offers an order of magnitude or more improvement in performance per unit cost compared to cluster-based scientific computers built from the same underlying semiconductor and packaging technology. We expect this efficiency to arise from two innovations: stream architecture and advanced interconnection networks. Organizing the computation into streams and exploiting the resulting locality using a register hierarchy enables a stream architecture to reduce the memory bandwidth required by representative computations by an order of magnitude or more.

Merrimac Node

- 90nm tech (1 V)
- ASIC technology
- 1 GHz (37 FO4)
- 128 GFLOPS

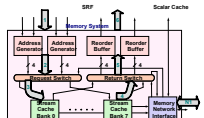


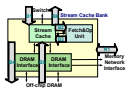


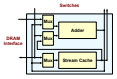
- Inter-cluster switch between clusters
- 127.5 mm2 (small ~12x10)
  - Stanford Imagine is 16mm x 16mm
  - MIT Raw is 18mm x 18mm
- 25 Watts per processor (P4 = 75 W)
- 41 Watts total per node (with DRAM)

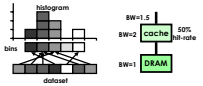
Memory System

- Single instruction accesses hundreds to thousands of words
  - Fill a very deep and wide memory pipeline
- High per-node bandwidth
  - 16 banks of RDRAM for 38.4 GB/s peak
  - Memory access scheduling
    - Improves average DRAM bandwidth
  - Bandwidth amplification through stream cache
- High bandwidth global memory space
  - Segment registers for translating virtual addresses
  - Network controller performs remote accesses
  - High radix routers allow for efficient single word messages
- Remote stream operations (scatter-add)
  - Increase data-parallel performance on "difficult" algorithms



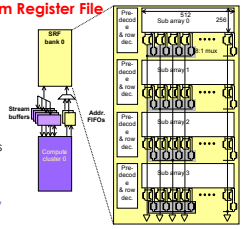






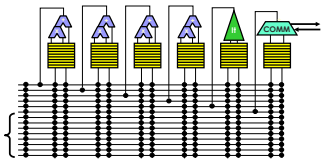
Stream Register File

- Single ported memory
  - Efficient wide access of 4 contiguous words
- Implemented using sub arrays
  - Reduced access time
  - Reduced power
- Stream-buffers match bandwidth to compute needs
  - Time multiplex the SRF port
- Indexed SRF at low extra cost
  - 8:1 MUX in sub-arrays
  - Row decoder per sub-array

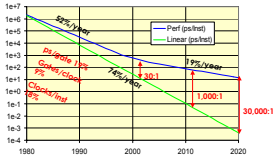


Compute Cluster

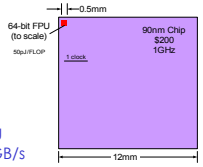
- 4 64-bit MADD units
  - Fully pipelined with 5 cycle latency
- Iterative operation support
  - Divide, sqrt, …
  - Acceleration unit
- Communication unit
  - Inter-cluster communication
- Distributed local register files
  - Low area and power



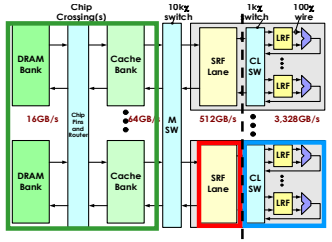
Requirements for Achieving High Performance on Modern Semiconductor Processes



- Parallelism
  - 100s FPUs per chip (millions per system)
- Latency Tolerance
  - 500 cycle remote memory access
- Locality
  - To match 20Tb/s ALU bandwidth to ~100GB/s chip bandwidth

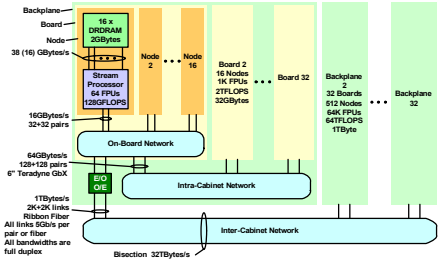


Bandwidth/Register Hierarchy

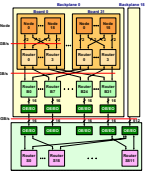


- Exploits locality
  - Producer-consumer within kernel in the LRF
  - Producer-consumer across kernels in the SRF
- Reduces the distance data travels
- Support large number of ALUs

Merrimac System

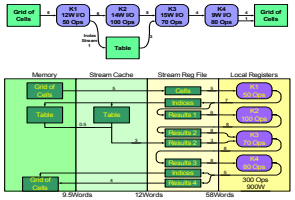


- Flat memory bandwidth within a 16-node board
- 4:1 Concentration within a 32-node backplane, 8:1 across a 32 backplane system
- Routers with bandwidth B=640Gb/s route messages with length L=128b
  - Requires high radix to exploit

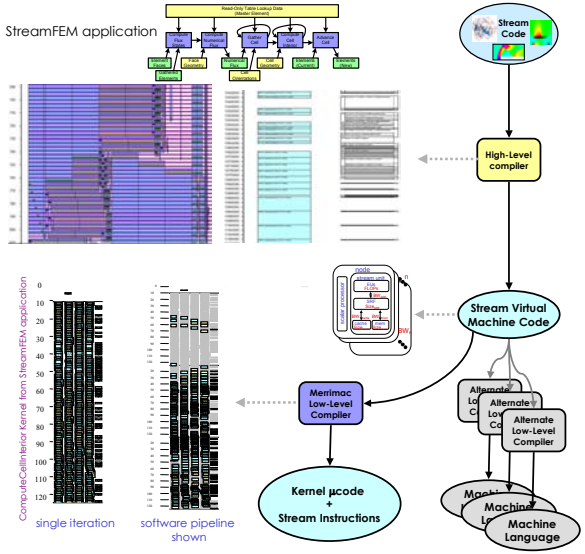


Stream Applications

- Locality
  - Producer-consumer within a kernel (local register file)
  - Producer-consumer across kernels (on-chip SRF)
  - Stream locality (spatial)
- Parallelism
  - Data Level Parallelism across stream elements
  - Task Parallelism across kernels



Compilation System



Application Results

Application	Sustained GFLOPS	FP Ops / Mem Ref	LRF Refs	SRF Refs	Mem Refs
StreamFEM3D[1] (Euler, quadratic)	31.6	17.1	153.0M (95.0%)	6.3M (3.9%)	1.8M (1.1%)
StreamFEM3D[1] (MHD, constant)	39.2	13.8	186.5M (99.4%)	7.7M (0.4%)	2.8M (0.2%)
StreamMD[1] (grid algorithm)	14.2[2]	12.1[2]	90.2M (97.5%)	1.6M (1.7%)	0.7M (0.8%)
GROMACS	38.8[2]	9.7[2]	108M (95.0%)	4.2M (3.7%)	1.5M (1.3%)
StreamFLO	12.9[2]	7.4[2]	234.3M (95.7%)	7.2M (2.9%)	3.4M (1.4%)

1. Simulated on a machine with 64GFLOPS peak performance
2. The low numbers are a result of many divide and square-root operations